IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	§	Group Art Unit: 2195
	§	
Grzegorz J. Czajkowski	§	Examiner: Wai, Eric Charles
Glenn C. Skinner	§	
Ciaran J. Bryce	§	Atty. Dkt. No.: 6000-33400
Stephen C. Hahn	§	
Peter James Soper	§	
	§	
Serial No.: 10/783,625	§	
	§	
	§	
Filed: February 20, 2004	§	
	§	
For: System and Method for	§	
Management of Characterized	§	
Resources	§	
	§	

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed November 18, 2008, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel/Frame 015014/0776, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III.   STATUS OF CLAIMS

Claims 1-45 are pending in the application and stand finally rejected.   The rejection of claims 1-45 is being appealed.   A copy of claims 1-45 is included in the Claims Appendix herein below.

IV.   STATUS OF AMENDMENTS

No amendments have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed toward a computer-readable storage medium storing program instructions. (*See, e.g.,* FIG. 13, elements 1307A-1307F; and p. 40, paragraph [1141].

The program instructions are computer-executable to perform operations including encoding an association of a computer resource and a resource management policy for the computer resource. (*See, e.g.,* FIG. 6, policies 605, 609, and 613 associated with resource A, and policy 605 associated with resource B; p. 3, paragraph [1008]; and p. 21, paragraph [1081].)

The program instructions are also computer-executable to perform binding one or more encapsulated computations to the encoding. (*See, e.g.,* FIG 2A, isolates: 201, 203, 205, computations: 207, 209, 211, etc.; FIG. 6, isolate 621 bound to policies 603 and 607, isolate 623 bound to policy 605, and isolate 625 bound to policy 609; FIG. 7, manager isolate 715 creates bindings at circle 2; pp. 21-22, paragraphs [1082-1084]; and p. 24, paragraph [1089], *bind()* method.)

The program instructions are also computer-executable to perform executing the one or more encapsulated computations in accordance with the resource management policy. (*See, e.g.,* p. 23, paragraph [1086]; and FIG. 7, elements 4a and 4b.)

Independent claim 13 is directed to a computer-implemented method that includes encoding an association of a computer resource with a resource management policy for the resource. (*See, e.g.,* FIG. 6, policies 605, 609, and 613 associated with resource A, and policy 605 associated with resource B; p. 3, paragraph [1008]; and p. 21, paragraph [1081].)

The method also includes binding one or more isolates to the encoding, wherein isolates include encapsulated one or more computations with state independent of other computations. (*See, e.g.,* FIG 2A, isolates: 201, 203, 205, computations: 207, 209, 211, etc.; FIG. 6, isolate 621 bound to policies 603 and 607, isolate 623 bound to policy 605, and isolate 625 bound to policy 609; FIG. 7, manager isolate 715 creates bindings at circle 2; pp. 21-22, paragraphs [1082-1084]; and p. 24, paragraph [1089], *bind()* method.)

The method further includes executing the one or more isolates in accordance with the resource management policy. (*See, e.g.,* p. 23, paragraph [1086]; and FIG. 7, elements 4a and 4b.)

Independent claim 24 is directed to a data structure encoded on one or more machine-readable storage media. (*See, e.g.,* FIG. 7, resource domain 706A-706F; and p. 23, paragraph [1085].)

The data structure includes a first field to indicate a computer resource. (*See, e.g.,* FIG. 7, resource domain 706A-706F, "Resource" field; and p. 23, paragraph [1085].)

The data structure includes a second field to indicate a resource management policy. (*See, e.g.,* FIG. 7, resource domain 706A-706F, "Policy Action(s) and Triggers" field; and p. 23, paragraph [1085].)

The data structure includes a third field to indicate availability of the computer resource. (*See, e.g.,* FIG. 7, resource domain 706A-706F, "Reservations" field; and p. 23, paragraph [1085].)

Independent claim 30 is directed to a computer-readable storage medium storing program instructions. (*See, e.g.,* FIG. 13, elements 1307A-1307F; and p. 40, paragraph [1141].

The program instructions are computer-executable to perform operations including preventing binding of two or more encapsulated computations with resource domain structures that indicate the same computer resource. (*See, e.g.*, p. 24, paragraph [1089].)

Each of the resource domain structures represents an association between a computer resource and a resource management policy. (*See, e.g.*, FIG. 6, policies 605, 609, and 613 associated with resource A, and policy 605 associated with resource B; p. 3, paragraph [1008]; and p. 21, paragraph [1081].)

The program instructions are also computer-executable to perform allowing binding of two or more encapsulated computations to resource domain structures that indicate different computer resources. (*See, e.g.*, FIG. 6, resource domains 601, 607, and 611; policies 605, 609, and 613 associated with resource A, and policy 605 associated with resource B; p. 3, paragraph [1008]; p. 21, paragraphs [1081-1083]; FIG. 6, isolate 621 bound to policies 603 and 607, isolate 623 bound to policy 605, and isolate 625 bound to policy 609.)

The program instructions are further computer-executable to perform executing the two or more bound encapsulated computations in accordance with the resource management policy. (*See, e.g.*, p. 23, paragraph [1086]; and FIG. 7, elements 4a and 4b.)

Independent claim 35 is directed to a computer-readable storage medium comprising program instructions. (*See, e.g.,* FIG. 13, elements 1307A-1307F; and p. 40, paragraph [1141].

The program instructions are computer-executable to implement instantiating an instance of a resource domain according to a resource domain class definition. (See, e.g., p. 3, paragraph [1008]; and p. 19, paragraph [1074].

The resource domain class definition provides for associating a computer resource with a resource management policy and for binding a set of one or more isolates to the instance. (*See, e.g.*, FIG. 6, resource domains 601, 607, and 611; policies 605, 609, and 613 associated with resource A, and policy 605 associated with resource B; p. 3, paragraph [1008]; p. 21, paragraphs [1081-1083]; FIG. 6, isolate 621 bound to policies 603 and 607, isolate 623 bound to policy 605, and isolate 625 bound to policy 609.)

Each of the isolates includes a set of one or more encapsulated computations with state independent of other isolates. (*See, e.g.,* FIG 2A, isolates: 201, 203, 205, computations: 207, 209, 211, etc.; and p. 5, paragraph [1026].)

The program instructions are also computer-executable to implement executing the one or more isolates in accordance with the resource management policy. (*See, e.g.*, p. 23, paragraph [1086]; and FIG. 7, elements 4a and 4b.)

Independent claim 42 is directed to an apparatus that includes a memory. (*See, e.g.*, FIG. 13, computer system 1300, including memory in the form of machine-readable media 1307A-1307F and storage device(s) 1309A-1309D; and p. 40, paragraph [1141].)

The apparatus also includes means for representing an association between a computer resource and a resource management policy and for binding one or more isolates with the representation of the association of the computer resource and the resource management policy. (*See, e.g.*, FIG. 6, resource domains 601, 607, and 611; policies 605, 609, and 613 associated with resource A, and policy 605 associated with resource B; p. 3, paragraph [1008]; p. 21, paragraphs [1081-1083]; FIG. 6, isolate 621 bound to policies 603 and 607, isolate 623 bound to policy 605, and isolate 625 bound to policy 609.)

An isolate includes a set of one or more computations with a state independent of other computations. (*See, e.g.,* FIG 2A, isolates: 201, 203, 205, computations: 207, 209, 211, etc.; and p. 5, paragraph [1026].)

The apparatus also includes means for executing the one or more isolates in accordance with the resource management policy. (*See, e.g.,* p. 23, paragraph [1086]; and FIG. 7, policy imposing isolate 717, and elements 4a and 4b.)

Dependent claim 45 is directed to the apparatus of claim 42, further including means for indicating usage of the computer resource. (*See, e.g.,* FIG. 7, resource domain 706A-706F, "Reservations" field; p. 23, paragraph [1085], and p. 25, [1091], getUsage() method.)

The summary above describes various examples and embodiments of the claimed subject matter; however, the claims are not necessarily limited to any of these examples and embodiments. The claims should be interpreted based on the wording of the respective claims.

VI.   GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1.   Claims 24-29 stand finally rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

2.   Claims 1-3, 11-23 and 42-45 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jones, et al. (U.S. Patent 6,003,061) (hereinafter "Jones").

3.   Claims 4-10 and 30-41 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Jones in view of Back, et al. ("Processes n KaffeOS: Isolation, Resource Management, and Sharing in Java") (hereinafter "Back").

4.   Claims 24-29 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Karch (U.S. Patent 7,096,219).

VII. ARGUMENT

First ground of rejection:

The Examiner rejected claims 24-29 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Appellants traverse the rejection of claims 24-29 for at least the following reasons.

The Examiner rejected claims 24-29 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Specifically, the Examiner submits that method claims and claims that recite a judicial exception (software) must recite a practical application, which may be provided by a physical transformation or a useful, concrete, and tangible result. Appellants assert that the claims, as currently amended, clearly recite a useful, concrete, and tangible result in the technological arts, e.g., encoding associations between computer resources and resource management policies and binding those encodings to computations (and/or isolates), causing the computations/isolates to be executed according to the resource management policies. Appellants assert that the management of computer resources for execution of computations/isolates is clearly a practical application in the technological arts.

Also, claim 24 recites "one or more machine-readable storage media" which refers to a particular physical article, not software *per se*.

The Examiner further submits that claims 24-29 are directed to a data structure comprising a mere arrangement of data (i.e. nonfunctional descriptive material). Appellants assert that the data structure of claims 24-29 could not be considered to be "nonfunctional descriptive material" by one of ordinary skill in the art in light of the plain language of the claims. For example, claim 24 recites, "A data structure encoded on one or more machine-readable storage media, the data structure comprising: a first field to indicate a computer resource; a second field to indicate a resource management policy; and a third field to indicate availability of the computer resource." The fields of the

claimed data structure do not describe <u>nonfunctional, abstract ideas,</u> but instead represent parameters of real computer resources and policies for managing those resources. Appellants' specification clearly describes that the values contained in these fields are used in <u>managing computer resources.</u> In addition, according to MPEP §2106.01 "...a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, <u>and is thus statutory.</u>" Therefore, Appellants assert that the claimed on one or more machine-readable storage media encoded with a data structure that defines the above-noted functional interrelationships is clearly statutory.

For at least the reasons above, Appellants respectfully request removal of the rejection of claims 24-29 under 35 U.S.C. § 101.

Second ground of rejection:

The Examiner rejected claims 1-3, 11-23 and 42-45 under 35 U.S.C. § 103(a) as being unpatentable over Jones et al. (U.S. Patent 6,003,061) (hereinafter "Jones"). Appellants traverse the rejection of these claims for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 12, 13, and 42:

1.   **The cited art clearly fails to teach or suggest** *encoding an association of a computer resource and a resource management policy for the computer resource,* **as recited in claim 1.**

On p. 3 of the Final Action, Examiner cites column 5, lines 42-46 and 57 as allegedly teaching creating an association of a computer resource and a resource management policy for the computer resource, submitting, "each resource is registered with the local resource planner." The Examiner admits that Jones does not explicitly

teach encoding an association, but submits that it would have been obvious that an association between the resources and policy is encoded. Appellants assert, however, that the cited passages teach only that the resource planner "applies a policy" to resources requests and not that such a policy is associated with a particular computer resource, as in Appellants' claim. In fact, other passages in Jones describe that a single, generic policy is applied to resource requests by the resource planner, as stated in column 14, lines 11-14, "In general, the resource planner implements the policy of the policy module it supports. This resource module may be changed and different embodiments of the present invention may employ different policy modules" (emphasis added). Jones later describes that the policy implemented by the resource planner may change, but the resource planner still only employs one policy at a time for managing all resources and resource requests. Since the resource planner in Jones implements only a single policy at a time for all resources and resource requests, there would be no reason to encode an association between this policy and any given resource, as suggested by the Examiner.

2.   **The cited art clearly fails to teach or suggest *binding one or more encapsulated computations to the encoding,* as recited in claim 1.**

The Examiner (in the Final Action, p. 3) cites column 5, lines 39-41 as allegedly teaching "resources are allocated to activities." The allocation of resources to activities has absolutely nothing to do with the above-referenced limitation, which recites binding computations to the encoding (i.e., to the encoded association between a computer resource and a resource management policy). This is clearly not taught or suggested by Jones.

3.   **The cited art clearly fails to teach or suggest *executing the one or more encapsulated computations in accordance with the resource management policy,* as recited in claim 1.**

In his remarks regarding this limitation (Final Action, p. 3), the Examiner merely submits that it is inherent that the activities are executed. Appellants assert, however,

that the claim does not just recite that the computations are executed, but that they are executed <u>in accordance with the resource management policy to which they are explicitly bound</u>. Since Jones does not teach the encoded association and the binding, the above-referenced limitation is clearly not inherent in Jones. Just because an activity may be executed in Jones, that does teach executing an encapsulated computation <u>in accordance with the resource management policy to which it is explicitly bound</u>, as recited in Appellants' claim.

4.   The Examiner has failed to provide a valid reason for modifying the cited reference.

As noted above, the Examiner submits that that it would have been obvious to one of ordinary skill in the art at the time of the invention that an association between the resources and policy is encoded. The Examiner further submits (in the Final Action, pp. 3-4), "The skilled artisan would realize the existence of a relationship between the resource and resource policy in Jones would need to be 'encoded' in some manner in order to exist as Jones is directed to computer software systems." However, as discussed above, such a relationship <u>does not exist</u> in Jones. Instead, the resource planner of Jones implements <u>a single policy at a time for all resources and resource requests</u>. Therefore, there would be absolutely no reason to encode an association between this policy and any given resource, as suggested by the Examiner, as this encoding (and even the suggested association itself) would serve no purpose in the system of Jones. The resource planner in Jones applies a policy to resource requests that does not depend on the resource being requested. Thus, the Examiner's assertion that a skilled artisan would realize that the existence of a relationship between the resource and resource policy in Jones needs to be encoded is **<u>completely unsupported by any evidence of record</u>**. In fact, since the resource planner applies a policy to resource requests that does not depend on the resource being requested, a modification that associates a particular resource with a respective policy would be counter to the principle of operation described in Jones. "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references

are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959).

For at least the reasons above, the rejection of claim 1 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 13 and 42 include limitations similar to those discussed above. Therefore, the arguments presented above apply with equal force to these claims, as well.

Claims 2 and 23:

1.   **The cited art fails to teach or suggest** *wherein the encapsulated computations correspond to a collaborative application,* **as recited in claim 2.**

On p. 4 of the Final Action, the Examiner cites col. 4, lines 56-60 of Jones as allegedly teaching this limitation, submitting, "distributed systems run collaborative applications." Appellants assert, however, that this passage says nothing about collaborative applications, nor do all distributed system necessarily run collaborative applications. Furthermore, even if the invention of Jones were installed in a distributed system running a collaborative application, Jones would does not teach that encapsulated computations of such a collaborative application are bound to an encoding of an association between a computer resource and a resource management policy for the computer resource, as required by Appellants' claims.

For at least the reasons above, the rejection of claim 2 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 23 includes limitations similar to those of claim 2 and was rejected for the same reasons. Therefore, the arguments presented above apply with equal force to this claim, as well.

Claim 3:

1. The cited art fails to teach or suggest *wherein an encapsulated computation has a state independent of other encapsulated computations*.

On p. 4 of the Final Action, the Examiner cites Jones, col. 5, lines 23-37 as allegedly teaching this limitation, without providing any remarks about how he believes this passage teaches the claimed invention. **Appellants note that per MPEP 707.07(d), the ground of rejection in an Examiner's Action should be "fully and clearly stated."** Since the rejection does not unambiguously specify the particular elements of the cited reference that the Examiner considers to be equivalent to the particular elements of Appellants' claim, Appellants assert that the Examiner's rejection is not "fully and clearly stated." The cited passage does not describe anything about a state of an encapsulated computation, or whether it is independent of the state of any other encapsulated computation. Instead, it describes the notion of an "activity" as "an abstraction that serves as a generalization of a running program and is the unit of abstraction to which resources are allocated and against which resource usage is charged." This passage describes that each activity is modeled by an activity object, and also describes, "Further, an activity may span address spaces and machines and may have multiple threads of control associated with it." **Appellants assert that this appears to describe <u>interdependent</u> processes and/or threads, rather than encapsulated computations having independent state, as required by claim 3.**

For at least the reasons above, the rejection of claim 3 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 11:

1. The cited art fails to teach or suggest *wherein said binding the one or more encapsulated computations to the encoding comprises indicating to each of the encapsulated computations the encoding*.

On p. 4 of the Final Action, the Examiner cites col. 5, lines 39-41 as allegedly teaching this limitation, submitting, "the resource planner tells the activity." This passage actually states, "The resource planner tells an activity <u>what amount of a resource,</u> if any, is reserved for use by the activity" (emphasis added). This clearly does not indicate to the activity the <u>encoding</u> of Appellants' claims (i.e., one that encodes an association between a computer resource and a resource management policy for the computer resource), as no such encoding is taught or suggested by Jones.

For at least the reasons above, the rejection of claim 11 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 14:

1.   **The cited art fails to teach or suggest** *wherein the encoding indicates the computer resource*.

On p. 5 of the Final Action, the Examiner again cites col. 5, lines 39-41 as allegedly teaching this limitation, but does not provide any remarks about how he believes this passage teaches the claimed invention. Since the rejection does not unambiguously specify the particular elements of the cited reference that the Examiner considers to be equivalent to the particular elements of Appellants' claim, Appellants assert that the Examiner's rejection is not "fully and clearly stated." Furthermore, as noted above, this passage states, "The resource planner tells an activity <u>what amount of a resource, if any, is reserved for use by the activity</u>" (emphasis added). This clearly does not indicate to the activity the <u>encoding</u> of Appellants' claims (i.e., one that encodes an association between a computer resource and a resource management policy for the computer resource), much less one that indicates the computer resource.

For at least the reasons above, the rejection of claim 14 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 15 and 43:

1.   **The cited art fails to teach or suggest** *wherein the encoding further indicates a set of one or more policy actions corresponding to the resource management policy, wherein execution of the set of policy actions causes a policy decision to be generated for the computer resource,* **as recited in claim 15.**

On p. 5 of the Final Action, the Examiner cites col. 5, lines 56-59 as allegedly teaching this limitation, but does not provide any remarks about how he believes this passage teaches the claimed invention.   Since the rejection does not unambiguously specify the particular elements of the cited reference that the Examiner considers to be equivalent to the particular elements of Appellants' claim, Appellants assert that the Examiner's rejection is not "fully and clearly stated." Furthermore, this passage merely states that the resource planner "applies a policy" to a request for resources by an activity and "determines whether the resources should be granted or not." It does not teach or suggest anything about the <u>encoding</u> of Appellants' claims, much that such an encoding indicates *a set of one or more policy actions corresponding to the resource management policy*, as required by Appellants' claim.

For at least the reasons above, the rejection of claim 15 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 43 includes limitations similar to those of claim 15 and was rejected for the same reasons. Therefore, the arguments presented above apply with equal force to this claim, as well.

Claim 16:

1. **The cited art fails to teach or suggest** *a dispenser isolate retrieving the set of policy actions from the encoding and executing the set of policy actions to invoke a policy imposing isolate.*

On p. 5 of the Final Action, the Examiner again cites col. 5, lines 56-59 as allegedly teaching these limitations, submitting, "a resource planner program does the execution." As noted above, this passage describes that the resource planner "applies a policy" to a request for resources by an activity and "determines whether the resources should be granted or not." It clearly does not teach or suggest anything about <u>retrieving a set of policy actions from an encoding</u> (i.e. one meeting the limitations recited therefore in Appellants' claims), as no such encoding is taught or suggested by Jones.

For at least the reasons above, the rejection of claim 16 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 17:

1. **The cited art fails to teach or suggest** *wherein the encoding also indicates availability of the computer resource.*

On p. 5 of the Final Action, the Examiner again cites col. 5, lines 56-59 as allegedly teaching these limitations, submitting, "availability is determined in view of pending reservations." Appellants assert that this general reference to "pending reservations" clearly does not teach or suggest <u>an encoding according to the limitations of Appellants' claims</u> that also indicates availability of the computer resource (i.e., the computer resource corresponding to the encoded association). Appellants again assert that no such encoding is taught or suggested by Jones, with or without this additional indication.

For at least the reasons above, the rejection of claim 17 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 18:

1.   The cited art fails to teach or suggest *wherein the encoding also indicates a reservation on the computer resource*.

On p. 5 of the Final Action, the Examiner cites col. 5, lines 53-55 as allegedly teaching this limitation, but does not provide any remarks about how he believes this passage teaches the claimed invention. Appellants assert that the Examiner's rejection is not "fully and clearly stated." Furthermore, this passage merely describes that an activity seeks to reserve resources that it needs before it uses them, and that an activity reserves resources by requesting a reservation from the resource planner. This has absolutely nothing to do with an encoding according to the limitations of Appellants' claims that also indicates a reservation on the computer resource. Appellants again assert that no such encoding is taught or suggested by Jones, with or without this additional indication.

For at least the reasons above, the rejection of claim 18 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 19 and 20:

1.   The cited art fails to teach or suggest *wherein the resource management policy is defined by a policy imposing isolate that installs the resource management policy in the encoding,* as recited in claim 19, or *wherein the bound isolates include the policy imposing isolate,* as recited in claim 20.

On p. 6 of the Final Action, the Examiner admits that Jones does not explicitly teach *wherein the resource management policy is defined by a policy imposing isolate that installs the resource management policy in the encoding.* The Examiner submits, "It

is well known in the art that operating systems and runtime environments such as Java have processes that perform resource management. It would have been obvious to one of ordinary skill in the art at the time of the invention that a process or thread that manages resources would impose the management policies." Appellants assert that the Examiner's general reference to operating systems and runtime environments that have processes that perform resource management clearly does not teach or suggest the specific limitations recited in claims 19 and 20 regarding the installation of a resource management policy in an encoding according to the recited limitations of such an encoding. **Nothing in the evidence of record teaches or suggests such an encoding, nor the installation of a resource management policy in such an encoding by a resource management process.**

For at least the reasons above, the rejection of claims 19 and 20 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 21:

1. **The cited art fails to teach or suggest** *indicating the encoding in a registry of resource management policy-computer resource association encodings*.

On p. 6 of the Final Action, the Examiner cites col. 5, lines 42-43 as allegedly teaching this limitation, stating, "resources are registered, therefore a registry exists." This passage merely states that each resource is registered with the resource planner. This general reference to the registration of resources clearly does not teach or suggest the specific limitations of the registry recited in claim 21, i.e., a registry of resource management policy-computer resources association encodings, as no such encodings are taught or suggested by Jones.

For at least the reasons above, the rejection of claim 21 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 22:

1.   The cited art fails to teach or suggest *characterizing the computer resource with generic attributes, and wherein the generic attributes comprise disposable, revocable, reservable, and bounded.*

The Examiner admits (in the Final Action, p. 6) that Jones fails to teach the limitations of claim 22, but submits, "However it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Jones to indicate the computer resource with generic attributes. One would be motivated by the desire to perform resource management more effectively during allocation." **The Examiner's remarks are completely unsupported by the cited art and amount to nothing but hindsight speculation.** There is clearly nothing in Jones that teaches or suggests that characterizing computer resources with generic attributes would improve the system of Jones at all. In fact, since the system of Jones treats all resources the same (i.e., the resource planner applies a single resource policy to all resources), it appears that Jones teaches away from the usefulness of characterizing resources at all, much less using the specific attributes recited in Appellants' claim.

For at least the reasons above, the rejection of claim 22 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 44:

1.   The cited art fails to teach or suggest *wherein the resource management policy further comprises triggers that gate execution of policy actions.*

The Examiner admits (in the Final Action, p. 7) that Jones fails to teach the limitations of claim 44, but submits, "However it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Jones to include triggers. It is well known in the art that management policies comprise rules based on inputs that

allow for determinations to be made on those inputs." **The Examiner's remarks are completely unsupported by the cited art and amount to nothing but hindsight speculation.** The Examiner's general reference to <u>management policies comprising rules</u> teaches nothing about the specific limitations of Appellants' claims involving resource management policies that are associated with computer resources and representations of those associations that are bound to one or more isolates.

For at least the reasons above, the rejection of claim 44 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 45:

1.   **The cited art fails to teach or suggest** *means for indicating usage of the* *computer resource.*

On p. 7 of the Final Action, the Examiner again cites col. 5, lines 56-59 as allegedly teaching these limitations, submitting, "availability is determined in view of pending reservations." Appellants assert that this general reference to "pending reservations" clearly does not teach or suggest <u>an encoding according to the limitations of</u> <u>Appellants' claims</u> that also indicates usage of the computer resource (i.e., the computer resource corresponding to the encoded association). Appellants again assert that no such encoding is taught or suggested by Jones, with or without this additional indication.

For at least the reasons above, the rejection of claim 45 is unsupported by the cited art and removal thereof is respectfully requested.

Third ground of rejection:

The Examiner rejected claims 4-10 and 30-41 under 35 U.S.C. § 103(a) as being unpatentable over Jones in view of Back, et al. ("Processes n KaffeOS: Isolation,

Resource Management, and Sharing in Java") (hereinafter "Back"). Appellants traverse the rejection of these claims for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claim 30:

1.   The cited art fails to teach or suggest *binding of two or more encapsulated computations to resource domain structures... and wherein each of the resource domain structures represents an association between a computer resource and a resource management policy.*

These limitations are similar to those discussed above regarding claim 1, and so the arguments presented above regarding claim 1 apply with equal force to this claim, as well.

2.   The cited art fails to teach or suggest *preventing binding of encapsulated computations with resource domain structures that indicate the same computer resources and allowing binding of computations with resources domain structures that indicate different computer resources.*

On p. 10 of the Final Action, the Examiner admits that Jones does not explicitly teach preventing the binding of two encapsulated computations with the same resource. The Examiner submits that Jones implies that different activities can be allocated to different local resources in view of pending reservations (column 5, lines 53-59). The Examiner submits that it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Jones to explicitly teach preventing the binding of activities to a particular resource because the skilled artisan would realize that Jones' teachings allow for activities to be allocated to different resources is a particular resource is unavailable. **Appellants assert, however, that the above-referenced limitations are not directed to whether resources are allocated to activities or computations, but to whether resource domain structures are bound to particular encapsulated**

computations. Thus, the Examiner's remarks are not relevant to what is actually recited in the claim. Also, the cited passages clearly do not teach the limitations actually recited in the claim. Column 5, lines 53-59, of Jones implies absolutely nothing about whether or not resource domain structures are bound to particular encapsulated computations. Nor is Back relevant to this aspect of Appellants' claim.

3.   The cited art fails to teach or suggest the resource domain structures of Appellants' claim.

The Examiner admits (in the Final Action, p. 10) that Jones does not explicitly teach the use of resource domains. The Examiner submits that Back teaches that the use of namespaces in Java allow processes to share or isolate resources (pg. 6 "3.2 Namespaces"). The Examiner submits that it would have been obvious to one of ordinary skill in the art that using namespaces in Java is equivalent to a resource domain structure since the resources are separated or shared depending on the situation. However, Appellants assert that the cited passages are directed to sharing of types, and a process/class loader that maps names to types. Appellants assert that this general description of the Java namespaces clearly does not teach or suggest the <u>resource domain structures</u> of Appellants' claims, *according to the specific limitations recited therein* (e.g., limitations regarding the <u>encoding of associations between particular computer resources and resource management policies</u> or limitations regarding the <u>criteria for determining whether or not</u> these structures are <u>bound to particular encapsulated computations</u>).

For at least the reasons above, the rejection of claim 30 is unsupported by the cited art and removal thereof is respectfully requested.

<u>**Claims 35, 37, and 38:**</u>

1.   The cited art clearly fails to teach or suggest *associating a computer resource with a resource management policy and… binding a set of one or more*

isolates to the instance... wherein each of the isolates includes a set of one or more encapsulated computations with state independent of other isolates; and executing the one or more isolates in accordance with the resource management policy.

These limitations are similar to those discussed above regarding claims 1 and 3, and so the arguments presented above regarding claims 1 and 3 apply with equal force to this claim, as well.

2.   The cited art fails to teach or suggest *instantiating an instance of a resource domain according to a resource domain class definition, wherein the resource domain class definition provides for associating a computer resource with a resource management policy and for binding a set of one or more isolates to the instance.*

On p. 11 of the Final Action, the Examiner admits that Jones does not explicitly teach the use of resource domains. The Examiner submits that Back teaches that the use of namespaces in Java allow processes to share or isolate resources (pg. 6 "3.2 Namespaces"). The Examiner submits that it would have been obvious to one of ordinary skill in the art that using namespaces in Java is equivalent to a resource domain structure since the resources are separated or shared depending on the situation. However, Appellants assert that the cited passages are directed to sharing of types, and a process/class loader that maps names to types. Appellants assert that this general description of the Java namespaces clearly does not teach or suggest the resource domain structures of Appellants' claims, according to the limitations recited therein (e.g., limitations regarding associating particular computer resources and resource management policies or limitations regarding the binding one or more isolates to particular instances of such a resource domain structure).

For at least the reasons above, the rejection of claim 35 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 4:

1. **The cited art fails to teach or suggest** *wherein said encoding the association includes instantiating a resource domain structure, wherein the resource domain structure indicates a computer resource.*

On p. 8 of the Final Action, the Examiner admits that Jones does not explicitly teach this limitation. The Examiner submits that Back teaches that the use of namespaces in Java allow processes to share or isolate resources (pg. 6 "3.2 Namespaces"). The Examiner submits that it would have been obvious to one of ordinary skill in the art that using namespaces in Java is equivalent to a resource domain structure since the resources are separated or shared depending on the situation. However, Appellants assert that the cited passages are directed to sharing of types, and a process/class loader that maps names to types. Appellants assert that this general description of the Java namespaces clearly does not teach or suggest *said encoding the association includes <u>instantiating a resource domain structure</u>,* as recited in claim 4. Appellants again assert that no such encoding is taught or suggested by the cited art, no matter how it is implemented.

For at least the reasons above, the rejection of claim 4 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 5:

1. **The cited art fails to teach or suggest** *wherein the encoding further includes indicating a set of one or more policy actions for the resource, the set of policy actions corresponding to the resource management policy.*

On p. 8 of the Final Action, the Examiner cites col. 5, lines 56-59 as allegedly teaching this limitation, but does not provide any remarks about how he believes this passage teaches the claimed invention. Since the rejection does not unambiguously specify the particular elements of the cited reference that the Examiner considers to be

equivalent to the particular elements of Appellants' claim, Appellants assert that the Examiner's rejection is not "fully and clearly stated." Furthermore, this passage merely states that the resource planner "applies a policy" to a request for resources by an activity and "determines whether the resources should be granted or not." It does not teach or suggest anything about the <u>encoding</u> of Appellants' claims, much that such an encoding indicates *a set of one or more policy actions corresponding to the resource management policy*, as required by Appellants' claim.

For at least the reasons above, the rejection of claim 5 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 6:

1.   **The cited art fails to teach or suggest** *wherein the program instructions are further executable to implement a policy imposing isolate installing the set of policy actions in the resource domain structure*.

On pp. 8-9 of the Final Action, the Examiner admits that Jones and Back do not explicitly teach these limitations. The Examiner submits, "It is well known in the art that operating systems and runtime environments such as Java have processes that perform resource management. It would have been obvious to one of ordinary skill in the art at the time of the invention that a process or thread that manages resources would impose the management policies." Appellants assert that the Examiner's <u>general reference</u> to operating systems and runtime environments that have processes that perform resource management clearly does not teach or suggest the <u>specific limitations</u> recited in claim 6 regarding the <u>installation</u> of a set of policy actions in a resource domain according to the recited limitations of such an encoding. **Nothing in the evidence of record teaches or suggests the resource domain structure of Appellants' claims nor the <u>installation of policy actions in such a structure</u> by a resource management process.**

For at least the reasons above, the rejection of claim 6 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 7:

1.   **The cited art fails to teach or suggest** *wherein the resource domain structure also indicates a set of one or more triggers for the resource, wherein the set of triggers correspond to respective ones of the set of policy actions.*

On p. 9 of the Final Action, the Examiner admits that Jones fails to teach the limitations of claim 7, but submits, "However it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Jones to include triggers. It is well known in the art that management policies comprise rules based on inputs that allow for determinations to be made on those inputs." **The Examiner's remarks are completely unsupported by the cited art and amount to nothing but hindsight speculation.** The Examiner's general reference to <u>management policies comprising rules</u> teaches nothing about the specific limitations of Appellants' claims involving policy actions that are associated with computer resources using resource domain structures that are bound to one or more encapsulated computations.

For at least the reasons above, the rejection of claim 7 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 8:

1.   **The cited art fails to teach or suggest** *wherein the resource domain structure also indicates a reservation on the resource.*

On p. 9 of the Final Action, the Examiner cites col. 5, lines 53-55 as allegedly teaching this limitation, but does not provide any remarks about how he believes this passage teaches the claimed invention. Appellants assert that the Examiner's rejection is

not "fully and clearly stated." Furthermore, this passage merely describes that an activity seeks to reserve resources that it needs before it uses them, and that an activity reserves resources by requesting a reservation from the resource planner. This has absolutely nothing to do with <u>a resource domain structure according to the limitations of Appellants'</u> <u>claims</u> that also <u>indicates a reservation</u> on the resource. Appellants again assert that no such resource domain is taught or suggested by the cited art, with or without this additional indication.

For at least the reasons above, the rejection of claim 8 is unsupported by the cited art and removal thereof is respectfully requested.

<u>Claim 9:</u>

1. **The cited art fails to teach or suggest** *wherein said binding the one or more encapsulated computations to the encoding comprises indicating in a registry each of the encapsulated computations and the encoding.*

On p. 9 of the Final Action, the Examiner cites col. 5, lines 42-43 of Jones as allegedly teaching this limitation, stating, "resources are registered, therefore a registry exists." This passage merely states that <u>each resource is registered</u> with the resource planner. This general reference to the registration of resources clearly does not teach or suggest the specific limitations of the registry recited in claim 9, i.e., <u>a registry that</u> <u>indicates each of the encapsulated computations and the encoding</u>, as no such encodings are taught or suggested by Jones.

For at least the reasons above, the rejection of claim 9 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 10:

1.   **The cited art fails to teach or suggest** *wherein the program instructions are further executable to implement a dispenser retrieving the policy actions from the resource domain structure and executing the policy actions to handle a resource request for the resource, wherein the dispenser is an isolate that handles requests for the resource.*

On pp. 9-10 of the Final Action, the Examiner again cites col. 5, lines 56-59 as allegedly teaching these limitations, submitting, "a resource planner program does the execution." As noted above, this passage describes that the resource planner "applies a policy" to a request for resources by an activity and "determines whether the resources should be granted or not." It clearly does not teach or suggest anything about retrieving policy action from a resource domain structure (i.e. one meeting the limitations recited therefore in Appellants' claims), as no such resource domain structure is taught or suggested by Jones.

For at least the reasons above, the rejection of claim 10 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 31:

1.   **The cited art fails to teach or suggest** *wherein the resource domain structures identify their resource domain and indicate resources and associated resource management policies.*

On p. 10 of the Final Action, the Examiner cites Jones (col. 5, lines 56-50) and Back (pg. 6, "Namespaces") as allegedly teaching these limitations, without providing any remarks about how he believes these passages teach or suggest them. Appellants assert that the Examiner's rejection is not "fully and clearly stated." Furthermore, as

discussed above, these passages clearly do not teach or suggest the <u>specific limitations</u> of the <u>resource domain structures</u> of Appellants' claims.

For at least the reasons above, the rejection of claim 31 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 32:

1.   **The cited art fails to teach or suggest** *wherein each of the resource domain structures indicate generic attributes of their computer resource that at least include disposable, revocable, reservable, and bounded.*

On p. 11 of the Final Action, the Examiner admits that Jones and Back fail to teach the limitations of claim 32, but submits, "However it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Jones to indicate the computer resource with generic attributes. One would be motivated by the desire to perform resource management more effectively during allocation." **The Examiner's remarks are completely unsupported by the cited art and amount to nothing but hindsight speculation.** There is nothing in the cited art that teaches or suggests that characterizing computer resources with generic attributes would improve the system of Jones/Back at all. In fact, since the system of Jones treats all resources the same (i.e., the resource planner applies a single resource policy to all resources), it appears that Jones <u>teaches away from</u> the usefulness of characterizing resources at all, much less using the generic attributes recited in Appellants' claim. In addition, since Jones/Back does not teach the resource domain structures of Appellants' claims, they clearly do not teach or suggest these additional limitations regarding <u>indications of these generic attributes</u>.

For at least the reasons above, the rejection of claim 32 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 33:

1. **The cited art fails to teach or suggest** *wherein the resource domain structures indicate usage of their computer resource*.

On p. 11 of the Final Action, the Examiner cites Jones, col. 5, lines 39-46, as allegedly teaching this limitation without providing any remarks about how he believes these passages teach or suggest them. Appellants assert that the Examiner's rejection is not "fully and clearly stated." Furthermore, this passage describes that the resource planner tells an activity what amount of a resource, if any, is reserved for use by the activity, and monitors what activities are allowed to gain access to a resource and how much of the resource may be granted to each activity. This passage describes nothing about a resource domain structure (according to the limitations of Appellants' claims) that also indicates usage of its computer resource. Appellants again assert that no such structure is taught or suggested by Jones and Back, with or without this additional limitation.

For at least the reasons above, the rejection of claim 33 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 34:

1. **The cited art fails to teach or suggest** *wherein the resource domain structures indicate reservations on their corresponding computer resource*.

On p. 11 of the Final Action, the Examiner cites col. 5, lines 56-59 as allegedly teaching this limitation, but does not provide any remarks about how he believes this passage teaches the claimed invention. Appellants assert that the Examiner's rejection is not "fully and clearly stated." This passage merely describes that the resource planner "applies a policy" to a request for resources by an activity and "determines whether the resources should be granted or not." This has absolutely nothing to do with a resource

domain structure according to the limitations of Appellants' claims that also indicates a reservation on a corresponding computer resource. Appellants again assert that no such structure is taught or suggested by Jones and Back, with or without this additional indication.

For at least the reasons above, the rejection of claim 34 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 36:

1.   **The cited art fails to teach or suggest *wherein the resource domain class definition provides a routine for determining current usage corresponding to an instance of the resource domain class*.**

On p. 12 of the Final Action, the Examiner cites Jones, col. 5, lines 22-23, as allegedly teaching this limitation, noting only, "resource accounting." This passage states, "...resource providers support operations such as allocating amounts of the resources, performing resource accounting, and providing notifications." This general reference to "resource accounting" clearly does not teach or suggest the specific limitations of Appellants' claim, including a resource domain class definition, or a routine for determining usage corresponding to an instance of a resource domain class. Appellants again assert that no resource domain class or structure (according to the limitations of Appellants' claims) is taught or suggested by Jones and Back, with or without the above-referenced limitation.

For at least the reasons above, the rejection of claim 36 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 39:

1.   **The cited art fails to teach or suggest** *wherein the program instructions are further executable to implement one or more routines for indicating computations bound to the resource domain class instance*.

On p. 12 of the Final Action, the Examiner cites col. 5, lines 59-60 of Jones as allegedly teaching the limitations of this claim without providing any remarks about how he believes these passages teach or suggest them. Appellants assert that the Examiner's rejection is not "fully and clearly stated." Furthermore, this passage merely states, "If the resources are granted to the activity, the activity may proceed to use the resources." Appellants assert that this does not teach or suggest anything about indicating computations that are bound to a resource domain class instance, since Jones and Back do not teach the resource domain class of Appellants' claims, or <u>binding</u> computations to an instance of such a class.

For at least the reasons above, the rejection of claim 39 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 40:

1.   **The cited art fails to teach or suggest** *wherein the program instructions are further executable to implement regulating association of computations with instances of the resource domain class, wherein each instance of the resource domain class indicates different resources*.

The Examiner admits (in the Final Action, p. 12) that Jones and Back do not teach the limitations of this claim. On p. 13 of the Final action, the Examiner submits, "It is well known in the art that Java is an object-oriented program that uses class definitions to construct multiple instances of an object. It would have been obvious to one of ordinary skill in the art at the time of the invention to include that each instance of the resource

domain class indicates a different resource." **The Examiner's remarks are completely unsupported by the cited art and amount to nothing but hindsight speculation.** There is nothing in the cited art that teaches or suggests the specific class definitions recited in Appellants' claims or how instances of a resource domain class (which is not taught by the references) differ from each other. In addition, Appellants' claim does not merely recite that each resource domain class instance indicates a different resource, but also recites program instructions executable to regulate the association of computations with those instances. Appellants assert that this is also not taught or suggested by the cited art or by the Examiner's remarks regarding generic class instances.

For at least the reasons above, the rejection of claim 40 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 41:

1. **The cited art fails to teach or suggest** *wherein the program instructions are further executable to implement associating resource domain class instances with dispensers that handle resource requests separately from implementation of the resource.*

On p. 13 of the Final Action, the Examiner cites Jones, col. 5, lines 38-41 as allegedly teaching the limitations of this claim, submitting, "resource planners are not involved with the implementation of the resource." Appellants assert that the Examiner's remarks (and the brief passage cited by the Examiner) do not teach or suggest anything about associating resource domain class instances with dispensers, as required by Appellants' claim. Appellants again assert that Jones and Back do not teach or suggest the resource domain class instances of Appellants' claims at all, much less the specific limitations of claim 41.

For at least the reasons above, the rejection of claim 41 is unsupported by the cited art and removal thereof is respectfully requested.

Fourth ground of rejection:

The Examiner rejected claims 24-29 under 35 U.S.C. § 103(a) as being unpatentable over Karch (U.S. Patent 7,096,219). Appellants traverse the rejection of these claims for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claim 24:

1.   **The cited art clearly fails to teach or suggest a *data structure comprising: a first field to indicate a computer resource; a second field to indicate a resource management policy; and a third field to indicate availability of the computer resource.***

The Examiner admits (on p. 13 of the Final Action) that Karch does not teach a data structure comprising the elements recited in claim 24. The Examiner submits that Karch teaches a resource usage analysis tool to perform reporting on resources, availability, and policies (in column 4, lines 5-12). This passage, however, does not describe reporting on policies (as suggested by the Examiner), but reporting and analysis of data management system resources and defining rules that control access to those resources. It also does not describe reporting on the availability of resources, as suggested by the Examiner, but describes reporting on actual usage of the resources. Therefore, Karch does not teach a resource usage analysis tool that reports on the three elements recited in claim 24.

On pp. 13-14 of the Final Action, the Examiner submits, "It would have been obvious to one of ordinary skill in the art at the time of the invention that Karch would also include a data structure to include all the fields. One would be motivated by the desire to allow for simplified reporting as provided by Karch (column 4, lines 17-20)."

Appellants assert that nothing in this passage of Karch describes simplified reporting, or a need for such simplified reporting. In addition, the Examiner has not provided any reason or evidence to support his assertion that the use of a data structure in Karch, (whether or not it included the fields recited in Appellants' claim) would allow for simplified reporting. Furthermore, Appellants' claim has nothing to do with reporting of computer resource usage. Therefore, the Examiner's reason to modify Karch is not commensurate with the feature he is attempting to include in Karch to result in the claimed invention. Therefore, the Examiner's reason to combine is improper. Finally, as discussed above, Karch does not teach the use of the specific elements recited in claim 24 in its resource analysis tool. Therefore, even if the elements described as being included in one of the reports generated by Karch were included in a single data structure, this data structure would not teach the data structure recited in Appellants' claim.

In the Response to Arguments section of the Final Action (p. 16) mailed September 25, 2008, the Examiner submits, "In response to applicant's argument that Karch does not describe reporting on the availability of resources, a recitation of the intended use of the claimed invention must result in a structural difference between the claimed invention and the prior art in order to patentably distinguish the claimed invention from the prior art. If the prior art structure is capable of performing the intended use, then it meets the claim." Appellants note that claim 24 does not merely recite an intended use, as the Examiner suggests. Instead, claim 24 recites <u>a data structure that includes three fields</u> **encoded** <u>on one or more machine-readable storage media</u> such that they indicate <u>three specific elements</u>. The Examiner has admitted that no such data structure is taught or suggested by Karch. Appellants assert, therefore, that the claimed invention clearly includes a structural difference over Karch, namely <u>a data structure specifically encoded on one or more machine-readable storage media in such a way that it indicates the recited elements</u>. **The Examiner has not cited anything in Karch that is analogous to this encoded data structure, or that is capable of providing the functionality of the recited three fields thereof.**

For at least the reasons above, the rejection of claim 24 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 25:

1. **The cited art fails to teach or suggest** *a fourth field to indicate an identifier to identify an association between a resource indicated in the first field and a resource management policy indicated in the second field.*

On p. 14 of the Final Action, the Examiner admits that the cited references do not teach the limitations of claim 25. The Examiner submits, "Karch does teach building aggregate tables and the addition of indexes can be performed (col 4 lines 17-20). It would have been obvious to one of ordinary skill to modify Karch to include a forth field to identify associations. One would be motivated to try to include such as association since it leads to predictable results." **The Examiner's remarks are completely unsupported by the cited art and amount to nothing but hindsight speculation.** The cited passage describes that analysis of various resource usage reports may provide recommendations about performance enhancements by <u>modifying the objects being managed</u> by the data management system (i.e., tables, files, etc.) **This has absolutely nothing to do with the limitations of Appellants' claim.** In addition, the Examiner has not provided any reason or evidence to support his assertion that the use of a data structure in Karch, (including the additional field recited in Appellants' claim 25) "leads to predictable results". As discussed above, there is nothing in Karch that teaches or suggests the <u>data structure</u> of Appellants' claims, much less one that includes <u>an identifier of an explicit association of a specific resource to a resource management policy</u>.

For at least the reasons above, the rejection of claim 25 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 26:

1.   The cited art fails to teach or suggest *a fourth field to indicate computer resource usage by a set of one or more encapsulated computations bound to the data structure*.

On p. 14 of the Final Action, the Examiner admits that the cited references do not teach the limitations of claim 26.   The Examiner submits, "Karch does teach the capability to report usage (col 4 lines 5-12).   It would have been obvious to one of ordinary skill to modify Karch to include a fourth field to indicate computer usage.   One would be motivated by the desire to track using resource usage on a per thread basis." **The Examiner's remarks are completely unsupported by the cited art and amount to nothing but hindsight speculation.**   The cited passage describes resource usage analysis and control segment 101, which provides, among other things, reporting of the data management system resources, in general.   There is nothing in Karch that describes the allocation of resources to computations, much less allocating resources on a <u>per thread basis</u>.   Therefore, the Examiner's suggestion that one would be motivated to track resource usage on a per thread basis is completely unfounded.   In addition, Karch does not teach or suggest <u>binding computations to a data structure</u> such as that of Appellants' claims.   Therefore, Karch clearly does not teach or suggest the use of an additional field in such a data structure to indicate resource usage <u>by a set of computations bound to the data structure</u>.

For at least the reasons above, the rejection of claim 26 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 27:

1.   The cited art fails to teach or suggest *wherein the first field indicates attributes of a computer resource*.

On p. 14 of the Final Action, the Examiner admits that the cited references do not teach the limitations of claim 27. The Examiner submits (Final Action, p. 15), "It would have been obvious to one of ordinary skill to modify Karch to uniquely identify each resource. It is well known in the art that attributes identify resources." **The Examiner's remarks are completely unsupported by the cited art and amount to nothing but hindsight speculation.** As discussed above, Karch does not teach the data structure of Appellants' claims. Therefore, Karch clearly does not teach or suggest the additional limitations of claim 27. Furthermore, Appellants' claim does not recite that the first field uniquely identifies each resource, as the Examiner suggests, but recites that the first field indicates <u>attributes</u> (plural) of a computer resource. The Examiner's <u>general reference</u> to the term "attributes" does not teach or suggest the specific use of a field that indicates <u>attributes</u> of a computer resource in the data structure of Appellants' claim to indicate a computer resource.

For at least the reasons above, the rejection of claim 27 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 28:

1.   **The cited art fails to teach or suggest** *wherein the attributes of the computer resource comprise: disposable, revocable, reservable, and bounded*.

On p. 15 of the Final action, the Examiner admits that Karch fails to teach the limitations of claim 28, but submits, "However it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Karch to indicate the computer resource with generic attributes. One would be motivated by the desire to perform resource management more effectively during allocation." **The Examiner's remarks are completely unsupported by the cited art and amount to nothing but hindsight speculation.** There is nothing in Karch that teaches or suggests that indicating computer resources with generic attributes, much less the <u>specific attributes</u> recited in Appellants' claim, would improve the system of Karch. In fact, Karch is not directed to

resource allocation, and does not mention resource allocation at all. Instead Karch is directed to a method of detecting fraud and/or unauthorized database access and enhancing the efficiency of a data warehouse system by utilizing a usage profile of a user. Therefore, the Examiner's remarks are completely unfounded.

For at least the reasons above, the rejection of claim 28 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 29:

1.   **The cited art fails to teach or suggest** *a fourth field to indicate a reservation of the computer resource*.

On p. 15 of the Final Action, the Examiner admits that the cited references do not teach the limitations of claim 26. The Examiner submits, "Karch does teach the capability to report usage (col 4 lines 5-12). It would have been obvious to one of ordinary skill to modify Karch to include a fourth field to indicate computer usage or reservation. One would be motivated by the desire to track using resource usage on a per thread basis." **The Examiner's remarks are completely unsupported by the cited art and amount to nothing but hindsight speculation.** The cited passage describes resource usage analysis and control segment 101, which provides, among other things, reporting of the data management system resources, in general. There is nothing in Karch that describes the allocation of resources to computations, much less allocating resources on a <u>per thread basis</u>. Therefore, the Examiner's suggestion that one would be motivated to track resource usage on a per thread basis is completely unfounded. **Furthermore, tracking resource usage on a per thread basis has nothing to do with the limitations of claim 29, which recites** *a fourth field to indicate <u>a reservation of the computer resource</u>*, **without any reference to a thread or any similar entity.** In addition, Karch does not teach or suggest the <u>data structure</u> of Appellants' claims. Therefore, Karch clearly does not teach or suggest the use of an additional field in such a data structure to indicate a <u>reservation of a computer resource indicated therein</u>.

For at least the reasons above, the rejection of claim 29 is unsupported by the cited art and removal thereof is respectfully requested.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-45 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/6000-33400/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: January 21, 2009

VIII.   CLAIMS APPENDIX

The claims on appeal are as follows.

1.   A computer-readable storage medium storing program instructions computer-executable to perform operations comprising:

encoding an association of a computer resource and a resource management policy for the computer resource;

binding one or more encapsulated computations to the encoding; and

executing the one or more encapsulated computations in accordance with the resource management policy.

2.   The computer-readable storage medium of claim 1, wherein the encapsulated computations correspond to a collaborative application.

3.   The computer-readable storage medium of claim 1, wherein an encapsulated computation has a state independent of other encapsulated computations.

4.   The computer-readable storage medium of claim 1, wherein said encoding the association includes instantiating a resource domain structure, wherein the resource domain structure indicates a computer resource.

5.   The computer-readable storage medium of claim 4, wherein the encoding further includes indicating a set of one or more policy actions for the resource, the set of policy actions corresponding to the resource management policy.

6.   The computer-readable storage medium of claim 5, wherein the program instructions are further executable to implement a policy imposing isolate installing the set of policy actions in the resource domain structure.

7. The computer-readable storage medium of claim 5, wherein the resource domain structure also indicates a set of one or more triggers for the resource, wherein the set of triggers correspond to respective ones of the set of policy actions.

8. The computer-readable storage medium of claim 4, wherein the resource domain structure also indicates a reservation on the resource.

9. The computer-readable storage medium of claim 4, wherein said binding the one or more encapsulated computations to the encoding comprises indicating in a registry each of the encapsulated computations and the encoding.

10. The computer-readable storage medium of claim 5, wherein the program instructions are further executable to implement a dispenser retrieving the policy actions from the resource domain structure and executing the policy actions to handle a resource request for the resource, wherein the dispenser is an isolate that handles requests for the resource.

11. The computer-readable storage medium of claim 1, wherein said binding the one or more encapsulated computations to the encoding comprises indicating to each of the encapsulated computations the encoding.

12. The computer-readable storage medium of claim 1, wherein the computer resource includes physical and logical computer resources.

13. A computer-implemented method, comprising:

   encoding an association of a computer resource with a resource management policy for the resource;

binding one or more isolates to the encoding, wherein isolates include encapsulated one or more computations with state independent of other computations; and

executing the one or more isolates in accordance with the resource management policy.

14. The method of claim 13, wherein the encoding indicates the computer resource.

15. The method of claim 14, wherein the encoding further indicates a set of one or more policy actions corresponding to the resource management policy, wherein execution of the set of policy actions causes a policy decision to be generated for the computer resource.

16. The method of claim 14, further comprising a dispenser isolate retrieving the set of policy actions from the encoding and executing the set of policy actions to invoke a policy imposing isolate.

17. The method of claim 14, wherein the encoding also indicates availability of the computer resource.

18. The method of claim 14, wherein the encoding also indicates a reservation on the computer resource.

19. The method of claim 14, wherein the resource management policy is defined by a policy imposing isolate that installs the resource management policy in the encoding.

20. The method of claim 19, wherein the bound isolates include the policy imposing isolate.

21. The method of claim 13, further comprising indicating the encoding in a registry of resource management policy-computer resource association encodings.

22. The method of claim 13 further comprising characterizing the computer resource with generic attributes, and wherein the generic attributes comprise disposable, revocable, reservable, and bounded.

23. The method of claim 13, wherein the isolates correspond to a collaborative application.

24. A data structure encoded on one or more machine-readable storage media, the data structure comprising:

a first field to indicate a computer resource;

a second field to indicate a resource management policy; and

a third field to indicate availability of the computer resource.

25. The data structure of claim 24 further comprising a fourth field to indicate an identifier to identify an association between a resource indicated in the first field and a resource management policy indicated in the second field.

26. The data structure of claim 24 further comprising a fourth field to indicate computer resource usage by a set of one or more encapsulated computations bound to the data structure.

27. The data structure of claim 24, wherein the first field indicates attributes of a computer resource.

28.   The data structure of claim 27, wherein the attributes of the computer resource comprise: disposable, revocable, reservable, and bounded.

29.   The data structure of claim 24 further comprising a fourth field to indicate a reservation of the computer resource.

30.   A computer-readable storage medium storing program instructions computer-executable to perform operations comprising:

preventing binding of two or more encapsulated computations with resource domain structures that indicate the same computer resource, wherein each of the resource domain structures represents an association between a computer resource and a resource management policy;

allowing binding of two or more encapsulated computations to resource domain structures that indicate different computer resources; and

executing the two or more bound encapsulated computations in accordance with the resource management policy.

31.   The computer-readable storage medium of claim 30 wherein the resource domain structures identify their resource domain and indicate resources and associated resource management policies.

32.   The computer-readable storage medium of claim 31, wherein each of the resource domain structures indicate generic attributes of their computer resource that at least include disposable, revocable, reservable, and bounded.

33.   The computer-readable storage medium of claim 31, wherein the resource domain structures indicate usage of their computer resource.

34.   The computer-readable storage medium of claim 31, wherein the resource domain structures indicate reservations on their corresponding computer resource.

35.   A computer-readable storage medium comprising program instructions computer-executable to implement:

    instantiating an instance of a resource domain according to a resource domain class definition, wherein the resource domain class definition provides for associating a computer resource with a resource management policy and for binding a set of one or more isolates to the instance, and wherein each of the isolates includes a set of one or more encapsulated computations with state independent of other isolates; and

    executing the one or more isolates in accordance with the resource management policy.

36.   The computer-readable storage medium of claim 35, wherein the resource domain class definition provides a routine for determining current usage corresponding to an instance of the resource domain class.

37.   The computer-readable storage medium of claim 35, wherein the program instructions are further executable to implement one or more routines for unconsuming computer resources.

38.   The computer-readable storage medium of claim 35 wherein the program instructions are further executable to implement one or more routines for attempting to consume a given amount of a computer resource, with the possibility of success or failure.

39.   The computer-readable storage medium of claim 35 wherein the program instructions are further executable to implement one or more routines for indicating computations bound to the resource domain class instance.

40. The computer-readable storage medium of claim 35 wherein the program instructions are further executable to implement regulating association of computations with instances of the resource domain class, wherein each instance of the resource domain class indicates different resources.

41. The computer-readable storage medium of claim 35 wherein the program instructions are further executable to implement associating resource domain class instances with dispensers that handle resource requests separately from implementation of the resource.

42. An apparatus, comprising:

a memory;

means for representing an association between a computer resource and a resource management policy and for binding one or more isolates with the representation of the association of the computer resource and the resource management policy, wherein an isolate includes a set of one or more computations with a state independent of other computations; and

means for executing the one or more isolates in accordance with the resource management policy.

43. The apparatus of claim 42, wherein the resource management policy comprises one or more policy actions that provide policy decisions to computer resource requests.

44. The apparatus of claim 43 wherein the resource management policy further comprises triggers that gate execution of policy actions.

45. The apparatus of claim 42 further comprising means for indicating usage of the computer resource.

IX.   <u>EVIDENCE APPENDIX</u>

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

X. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.